

**METHOD AND APPARATUS FOR GENERATING NORMALIZED  
REPRESENTATIONS OF STRINGS**

**BACKGROUND OF THE INVENTION**

5    **1. Field of the Invention**

The invention generally relates to a method and apparatus for generating normalized representations of strings, e.g. sentences, and in particular to a method for providing translation information for translating a string from a first language to a second language.

10   **2. Description of the related art**

09738349-121800  
A plurality of applications make use of normalized representations of strings, e.g. sentences, in particular applications in the area of translation memory, authoring memory, bilingual authoring memory, indexing, etc. An important application of normalized representations is translation memories in translation systems. These translation memories store linguistically-based normalized representations of text. Translation memory repositories collect segments of texts such as sentences or technical terms associated with a translation into some target languages. Such repositories give human translators an immediate access to translations that have been previously recorded. This reduces the effort, time and costs of translations, while improving its consistency.

The capability of translation memories can be expanded through fuzzy matching, a technique that matches input segments yet to be translated with segments stored in the translation memory, even if they are not identical to the input segments. Typical measures to allow fuzzy matching during the matching process may be ignoring a predefined set of words such as articles, conjunctions, etc or ignoring a set of predefined symbols, in particular punctuation marks. Furthermore, upper case and lower case characters or specific expressions such as numerical expressions may be normalized. During a matching step of a retrieving processing string segments (as ordered sequences of characters) may be used regardless of their linguistic structure and a mismatch of a certain number of characters may be allowed.

Retrieval systems serve to retrieve those texts or text portions that are relevant to the information needs of a user. In general, the relevant information contained in texts is constructed and extracted according to a normalized representation. Such a representation is abstracted away from its original linguistic form. Database queries of a user are generally subjected to processing in order to expand the scope of the query and/or to interpret the query syntax. The extracted query information is then matched against the stored representations in order to retrieve the specific information contained in a text. That or those text units which are the most similar to a query are output as retrieved text units.

For evaluating retrieval performance of information retrieval systems, two criteria are used, namely the "calling rate" and the "precision" these criteria are based on the subjective point of view on the relevance of retrieved information. The "calling rate" or "recall" and the precision are defined as follows.

The recall is a ratio of a number of relevant retrieved text units to the total number of relevant text units stored in the database, the precision is a ratio of a number of relevant text units to the number of retrieved text units. There is usually a trade-off between these two criteria. In information retrieval, it is desirable that these two criteria are in proximity to the maximum value of one.

### SUMMARY OF THE INVENTION

The present invention has been made in consideration of the above situation, and it has as its primary object to provide a method and apparatus for generating normalized representations of strings.

It is another object of the present invention to provide a method and an apparatus that increase the proportion of relevant text units retrieved from a database.

It is still another object of the present invention to provide a method and an apparatus that retrieve information from a database with a higher precision.

It is yet another object of the invention to provide a method and an apparatus for generating normalized representations which may be stored in a database wherein the normalized representations increase the retrieval performance of information retrieval systems.

It is another object to provide a translation memory for translation systems having an increased retrieval performance of translation information.

Still another object of the present invention is to provide a method for providing translation information with improved retrieval performance from  
5 previously translated text units.

These and other objects of the present invention will become apparent hereinafter.

To achieve these objects, the present invention provides a method, and apparatus therefor, for normalizing input strings. The method includes receiving  
10 the input strings for linguistic analysis, which includes morphological, syntactic and semantic analysis, to generate a first representation of each of the input strings. Each of the first representations includes linguistic information that relates, for example, signs such as sounds and words to their meaning and vice versa. Each of the first representations is skeletised to generate a  
15 corresponding second representation for each of the input strings. The skeletising replaces the linguistic information with abstract variables in each of the second representations. The second representation are then stored as normalized representations of the input strings.

According to another aspect of the invention, there is provided a retrieval  
20 method for retrieving strings from a database. The database comprises normalized representations of a plurality of strings. The retrieval method comprises the steps of receiving a query, generating a normalized representation of the query, matching the normalized representation of the query to the normalized representations stored in the database and outputting  
25 the most similar string retrieved from the database.

According to a further aspect, there is provided a method for providing translation information for translating a string of a first language to a second language. The translation information is based on a plurality of normalized representations of strings in the first language and corresponding strings  
30 previously translated to the second language. The method for providing translation information comprising the steps of receiving an input string in the first language, retrieving a similar string in the first language from the plurality of

normalized representations and outputting the translation information based on a string in the second language which corresponds to the retrieved string in the first language.

5 In preferred embodiments, further improvements can be achieved by applying additional processing steps to the processing sequence, either by providing at least an operation out of the plurality of operating functions extracting linguistic information or by operating functions generalizing extracted linguistic information.

10 The invention has its primary applications in indexing, information retrieval, translation memory, authoring memory and machine edited human translation.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

15 The accompanying drawings are incorporated into and form a part of the specification to illustrate several embodiments of the present invention. These drawings together with the description serve to explain the principles of the invention. The drawings are only for the purpose of illustrating preferred and alternative examples of how the invention can be made and used and are not to be construed as limiting the invention to only the illustrated and described  
20 embodiments. Further features and advantages will become apparent from the following and more particular description of the various embodiments of the invention, as illustrated in the accompanying drawings wherein:

Fig. 1 illustrates a schematic diagram of a configuration of an information retrieval system;

25 Fig. 2 illustrates a schematic diagram of a computer system;

Fig. 3 is a flow-chart illustrating an information retrieval method;

Fig. 4 is a flow-chart illustrating in more detail a text unit processing step of the process as depicted in Fig. 3; and

30 Fig. 5 is an example of a binary tree used to represent different skeletisation functions.

## DETAILED DESCRIPTION

The illustrated embodiments of the present invention will now be described with reference to the figure drawings.

5 Referring now to the drawings and particularly Fig. 1, which illustrates a preferred embodiment of an information retrieval system 100. Such an information retrieval system might be a method for providing translation information. In the retrieval system, there is provided a "storage line" including a text-processing unit 101 and a unit 102 for storing normalized representations of  
10 text units which are received from processing unit 101. Such a unit 102 might be a text unit database, in particular a translation memory in a translation system as a foreign language aid.

Many databases use indexes for the most frequently requested information contained therein. In other words, each field of information for each  
15 record in the database is indexed. An index can be described as a list of distinct attribute record values associated with pointers. By indexing individual fields of a database, queries on index fields enjoy a fast response time. In a preferred embodiment, the processing unit 101 creates normalized representations of a plurality of text units as indexes for these text units.

20 The processing of input strings or text units in unit 101 is based on a plurality of text analyzing functions. The present invention proposes a representation scheme for storing, matching, retrieving text units. A sequence of processes is used to produce various representations. The present invention uses natural language processing capabilities such as part-of-speech tagging,  
25 lemmatization and syntactic or semantic notations, skeletisation and any combination of such means. Normalizing is any process used to produce such representations.

For describing such normalization, the following sentence will be used as an example:

30 *When you are working with the HomeCenter you may  
need to use the buttons to do one or more tasks.*

This example sentence may be normalized and represented through different levels of normalization which may result from one of the processes described in the following.

- As a first normalization process applied to the example sentence
- 5 morphological variance of individual words is normalized according to its base form. An example of normalizing a word to its base form is as follows.

*"working" is normalized to "work",*

*"buttons" is normalized to "button".*

- In addition, a syntactic category preferably disambiguated, is assigned to
- 10 each word. The syntactic category is determined based on a context analysis. The following examples may be assigned to individual words of the above example:

*"use" is recognized as a "verb"*

*"HomeCenter" is recognized as a "proper noun",*

- 15 *"buttons" is recognized as a "plural noun".*

- Based on the above normalizing operations, particular linguistic units, so-called tags, are assigned to each word. Such "part-off-speech" tags may be +VB to indicate the syntactic category "verb". As an example, "use " would be tagged by +VB, "HomeCenter" would be tagged as +PN for a proper noun and "buttons"
- 20 would be tagged by +NNS for being a plural noun.



“when”. The beginning and end of the “when”-sub-clause is indicated by markers, e.g. “SC\_WHEN”. The infinitive clause is enclosed by markers “IV”. The above example additionally shows some dependency relations which will be described in the following in detail.

5        A syntactic analysis may also provide explicit dependency relations between sub-strings extracted for selected predefined relations. As shown in the above syntactic information representing the first example sentence, “you” is indicated as being the subject of the verb “work”. Further, the term “button” is indicated as being the object of the word “use”. The expression “task” is  
10 indicated to be the object of the word “do”. The expression “HomeCenter” is indicated to modify the verb “work”. Such dependency relations may also be noted as follows:

*SUBJ(you, work)*

*SUBJ(you, need)*

15        *OBJ(use, button)*

*OBJ(do, task)*

*VMODOBJ(work, with, HomeCenter)*

Such relations may be used to find matches between common syntactic structures of sentences without regarding their actual surface realization. An  
20 example for a successful matching process for different sentences may be as follows:

*(a) The parts must be replaced by a qualified technician.*

*(b) Only qualified technicians may replace the parts.*

In both above example sentences “technician(s)” is the subject of the  
25 verb “replace”; the term “parts” is the object of the verb “replace”.

For normalizing strings, e.g. sentence, also taxonomic and semantic links may be used. Such a semantic link may create from an individual word a connection to a super-ordinated term having a more general meaning, to a synonym word or even to another more specialized term for the same super-



ordinated term. Such semantic links may exist between "jet plane" and "civil aircraft" or between "cattle", "breeder" and "farmer".

Semantic normalization relies on pre-existing language resources such as ontologies, thesauri, terminological databases and on semantic disambiguation processes. By employing operations for generating semantic links, the first example sentence described above might be annotated with the following semantic tags:

*When you are working [WORK\_PROCESS] with the HomeCenter [MACHINE] you may need to use the buttons [PARTS] do one or more tasks.*

The above-described linguistic processes correspond to different levels of abstraction and provide information of increasing complexity. In addition, their representation scheme of the present invention makes use of another type of operation, named "skeletisation". Skeletisation operates on the output of linguistic processes such as described above and provides another level of abstraction. More precisely, skeletisation replaces selected linguistic units by abstract variables.

As an example, skeletisation operating on the output of a shallow parser and abstracting over direct objects, proper names and technical terms, produces the following normalized representation for the first example sentence:

*When you are working with <PROPER1>, you may need to use <OBJ1> to do <OBJ2>.*

In the above example, the two direct objects "the buttons" and "one or more tasks" are replaced by the index variables OBJ1 and OBJ2. Further, the expression "HomeCenter" is replaced by the variable PROPER1.

Different skeletisation schemes may be used which are based on various linguistic parameters. As another example adverbial clauses may be abstracted. Such a clause is the "when-clause" in the first example sentence. Such a skeletisation process would result in the following representation of the sub-sentence:

*<WHEN\_CLAUSE> you may need to use the buttons to do one or more tasks.*

In addition to the different possible kinds of skeletisation operations, it is possible to combine the operations into a complex one. Both above proposed skeletisation operations may be combined to produce the following representation:

5           <WHEN\_CLAUSE> *you may need to use* <OBJ1> to do <OBJ2>.

A skeletisation may be more generally seen as a function  $Skel(x, Abstr, Filter(x))$ . In this function, "x" is a linguistic representation attached to an input string, "Abstr" is the set of linguistic units to be abstracted in "x" and how they should be represented in the output, and "Filter(x)" specifies how not-abstracted elements of x should be re-written (e.g., translation). By default, "Filter(x)" produces a sequence of not-abstracted surface forms as shown in the example representation of both skeletisation functions.

The above described skeletisation processes may be defined by such skeletisation functions. The first described skeletisation operation is directed to abstract over direct objects, proper names and technical terms. This skeletisation function may be described by function  $Skel(x, \{OBJ, PN\},)$ . This function indicates that parts of a sentence marked by OBJ or PN must be abstracted, while the other elements of the sentence remain unchanged.

The second above described skeletisation process may be described by the following skeletisation function  $Skel(x, \{WHEN\_CLAUSE\},)$ . The finally described skeletisation procedure combining the first two may be described as  $Skel(x, \{OBJ, PN, WHEN\_CLAUSE\},)$ .

A linguistic unit to be abstracted may be specified in more detail in order to only abstract particular selected kinds of words. As an example, only female objects may be abstracted. Such a skeletisation function would have a parameter specifying the abstract parameter as OBJ&Fem. The parameter "Abstr" directly specifies the type of variable to replace the term in the sentence. Each such variable is indexed according to its position of occurrence in a string or sentence. Thus, the first direct object is marked by OBJ1 in order to uniquely identify this variable.

This skeletisation process highly depends on the linguistic representation in use. The examples above are only meant to illustrate the scope of invention.

A plurality of different skeletisation functions is obvious for a person skilled in the art based on the above given examples.

When performing a plurality of normalization functions, each results in a different representation for the input sentence. The normalization procedures may also be combined in different ways. Each input sentence can be stored together with a set of representations provided by the employed normalization procedures. The first example sentence may be stored together with the following sequence of three representations:

**1. Baseform and part-of-speech tagging:**

*When+WRB you+PPSS be+BER work+VBG with+IN the +AT  
Homecentre+NP you+PPSS may+MD need+VB to+TO use+VB the+AT  
button+NNS to+TO do+DO one+CD or+CC more+AP task+NNS .+SENT*

**2. Abstraction over When\_clauses:**

*<WHEN\_CLAUSE> you may need to use the buttons to do one or more tasks.*

**3. Abstraction over When\_clauses and objects:**

*<WHEN\_CLAUSE> you may need to use <OBJ1> to do <OBJ2>.*

As already mentioned the given sentence may result in many different sequences of representations depending on the different linguistic processes and skeletisation functions used. In the following the selection of appropriate skeletisation functions or sequences of skeletisation functions will be described.

In the form of  $\text{Skel}(x, \text{Abstr}, \text{Filter}(x))$ , "Abstr" defines those elements within a string, namely a sentence  $x$ , which are to be abstracted, i.e. replaced by a variable. Elements to be replaced by variables may be lexical units, phrases, clauses, or any text element which should be abstracted.

In the following, the term "skeleton" will be used for the result of a skeletisation function.

Skeletons may be ranked depending on their degree of generalization. When assuming  $\text{Skel1}=\text{Skel}(x1, \text{Abstr1}, \text{Filter}(x))$  and  $\text{Skel2}=\text{Skel}(x2, \text{Abstr2}, \text{Filter}(x))$ , then Skel1 is less general than Skel2. Those representations

produced by Skel2 include elements to be replaced according to Skel1. An example for such a different generalization degree is shown by the following example wherein skeleton (a) is more general than skeleton (b) as (b) may be produced from (a):

- 5           (a) WHEN\_CLAUSE *you may need* INF\_CLAUSE1 INF\_CLAUSE2  
          (b) WHEN\_CLAUSE *you may need to use* NP1 *to do* NP2

Skeletisation functions may be ranked according to their degree of generalization based on the following ordering property. For any non empty sets "Abstr1" and "Abstr2" of elements to be abstracted, in case  $Abstr1 \subseteq Abstr2$ ,  
10 then all skeletons obtained with Skel1 are less general than those skeletons obtained with Skel2. Then, the skeletisation function Skel1 is less general than the skeletisation Skel2.

Different schemes may be used to represent and select sequences of normalized representations efficiently. The following proposal for a  
15 representation in the form of a binary tree is a particular embodiment of the present invention. However, it is obvious to a person skilled in the art that a plurality of other representations may be possible based on the given examples.

When using the above ordering property, sequences of skeletisation functions may be represented using a binary tree. Each node of such a binary  
20 tree represents a skeletisation function. Each non-terminal node in the binary tree branches to two descendents, namely a left child and a right child. The left child represents the application of a non-empty skeletisation function, whereas its right child represents the identity function "idty". This identity function "idty" leaves the input unchanged. Thus, the identity function can be regarded as an  
25 empty skeletisation function (i.e. a skeletisation in which Abstr is empty).

A sequence of subsequently performed skeletisation functions is represented by a path within the binary tree. Such a path starts from the top of a tree and connects via a plurality of nodes (skeletisation operations) a terminal node.

30           An example will be given in connection with Fig. 5. Assuming that the following two sentences are subjected to sequences of skeletisation operations:

(1) *When you are working with the HomeCenter you may need to use the buttons to do one or more tasks.*

(2) *When you are working with the HomeCenter you may need the manual to do one or more tasks.*

5 The following elements are to be replaced by variables:

WHEN\_CLAUSE

INF\_CLAUSE (infinitive clause),

NP (non-plural)

VINF (verb, infinitive)

10 Based on the above listed skeletisation operations for the two example sentences, the binary tree of Fig. 5 may represent sequences of skeletisation functions. For sake of clarity, only a part of the tree is shown in Fig. 5.

As both sentences, namely sentence 1 and 2, may be abstracted using the whole set of above described skeletisation functions, they appear on the left  
15 most node of the tree. Such nodes may be marked in order to indicate that it is not possible to further abstract these representations based on a given set of seketisation functions.

For any given node N within the binary tree, those nodes which are located above or on the right of the given node N wherein the depth thereof is  
20 less than or equal to the one of N defines a set of nodes which are equivalent to N.

In a more particular embodiment, the set of text elements to be abstracted may be divided into two parts, the first part consisting of those elements which have to appear in the sentence in a precise order, and a second  
25 part consisting of those elements which may appear in any position within a sentence.

The above described binary tree representation is only one among others which may be derived from the described ordering property.

The selection of skeletons for sentences depends on the application for  
30 which the normalization procedure is intended. Skeletons relevant for translation

memories may not be relevant for retrieval or for monolingual authoring. For retrieval applications, it is important that a retained skeletisation functions partitions efficiently the set of sentences, so that the search for similar sentences involves few steps. Thus, those skeletisation functions are to be selected, which discriminate best-input sentences and which stop skeletising sentences when the associated skeletisation function applies only to a few sentences. Similar considerations have to be carried out for other kinds of applications. A selection of operating functions, in particular skeletisation functions, and their order may be carried out by machine learning.

Referring now again Fig. 1, the configuration of an information retrieval system 101 further comprises a "search line" which comprises a query formatting unit 104 for formatting a query and generating a normalized representation for the query. In the formatting stage, the query is transformed by the unit 104 into a normalized representation, using the same methods 103 as described for text units. The results of retrieving operations performed by a matching unit 106 will be output as retrieved strings.

As illustrated in Fig. 2, a computer system 200 according to the present invention comprises a processor 201 and a main storage 202 connected to the processor 201. The processor is further connected to an input unit 203 for inputting a data base query by a user. The results of the query are displayed on a display unit 204 or may be printed out via printer 205. The information is retrieved from a database 206 included in the computer system 200. The database may be provided on a hard disk of the computer system or on a particular storage media as CDs, DVDs or tape recording media. Alternatively, database 206 may be stored on a remote location connected to the computer system via a data transmission network. In a further alternative embodiment, database 206 may comprise texts stored in a plurality of computers connected in a network at high speed.

Processor 201 processing a query may also be used to process sentences. The processor 201 may further use a thesaurus memory 207. The thesaurus memory may provide semantic relations between terms or a plurality of terms. All those skilled in the art will appreciate, that the application of the method and apparatus according to the present invention is not limited to

particular memories. All parts of the memories 206, 207, may be realized as a distinct area in a single memory. As those skilled in the art will further appreciate, the computer system 200 preferably comprises a main memory 202 for storing program code, such as operating and application programs 101, 104, and 106 and data 102, 103, and 105. It will also be appreciated by those skilled in the art that the processor 201 executes program code to process strings or queries. The computer 200 preferably comprises an external memory, such as a hard disk drive and floppy disk drive, for storing a problem code and data memory permanently.

Computer system 200 may further comprise a parsing unit 208 in order to extract syntactic relations between terms of a sentence. Alternatively, parsing unit 208 may be provided as an application program to be performed by processor 201.

A process of information retrieval will now be described in more detail with reference to the flow-chart diagrams as depicted in Fig. 3 and Fig. 4.

Fig. 3 illustrates general steps in an information retrieval system. First, in a string processing step 301 normalized representations are generated for all strings or text units in a database. Details of the string processing are given in Fig. 4.

In order to retrieve strings from a database, the user has to input a database query to find those strings relevant to the given input information. The received query is processed in the computer system 200 in step 302 in order to be in a format which may be compared to the stored data in the database.

For a retrieval, the query has to be compared with the stored data. The retrieval of strings in step 303 is based on the similarity of each string with respect to the database query. The retrieved information, which may include a translated version, is presented in step 304 to a user based on the relevance of each string with respect to the database query.

Fig. 4 illustrates in more detail the processing of strings, i.e. text units or sentences, for obtaining a database according to the present invention. The manner of knowledge representation, namely architecture of the database is an essential aspect of the present invention. All those skilled in the art will

appreciate, that the particular knowledge representation and method of generating such knowledge representation is not restricted to a document retrieval application but may be a basis for knowledge extraction, translation information apparatus, etc.

5 First, in step 401 linguistic information is extracted from an input sentence as described above. Linguistic information includes morphological, syntactical, and semantic information and ways to disambiguate between. Based on the extracted information, a normalized representation is generated.

10 Such a normalizing procedure may be carried out with respect to different aspects a plurality of times. The performed sequence of operations will result in a corresponding plurality of normalized representations having a different degree of generalization.

15 Further, as indicated in step 402, a skeletisation operation is applied to the normalized representation(s). The skeletisation operations replace particular elements of a string or sentence by a variable in order to allow for better matches during an information retrieval process. Such variables will match to corresponding variables of other sentences regardless of the particular originally used text elements in text units or sentences to be matched.

20 The normalized representations resulting from the linguistic extracting step and the skeletisation step may each (or only selected ones) be stored together with the particular sentence, as for example strings 102.

Based on such databases, information retrieval and the provision of translation information may be improved based on an improved access to the relevant information stored.

25 It will be appreciated that portions of the system embodying the present invention may be readily implemented in software using software development environments that provide source code that can be used on a variety of hardware platforms. Alternatively, portions of the system may be implemented partially or fully in hardware using standard logic circuits. Whether software or  
30 hardware is used to implement different portions of the system varies depending on the speed and efficiency requirements of the system and also the particular



function and the particular software or hardware systems and the particular microprocessor or microcomputer systems being utilized.

Also it will be appreciated by those skilled in the art that any resulting program(s) incorporating the present invention, having computer-readable program code, may be embodied within one or more computer-usable media such as memory devices or transmitting devices, thereby making a computer program product or article of manufacture. As such, the terms "article of manufacture" and "computer program product" as used herein are intended to encompass a computer program existent (permanently, temporarily, or transitorily) on any computer-usable medium such as on any memory device or in any transmitting device.

00374 0733250